ADSP-2100 Family Development Software

ADDS-21XX-SOFTWARE

FEATURES

SYSTEM BUILDER
- Architecture Description File Specifies System Hardware

ASSEMBLER
- Easy-to-Program Algebraic Syntax
- Preprocessor Supports C Language Constructs
- Flexible Macro Processing
- Encourages Modular Code Development

LINKER
- Maps Assembler Output to Target System Memory
- Supports User-Defined Library Routines
- Generates Memory Map Listing

PROM SPLITTER/HIP SPLITTER
- Formats Executable File for Programming PROMs or for Host Processor Booting

SIMULATORS
- Reconfigurable, Windowed Interface
- Pull-Down Menus
- Full Symbolic Disassembly and On-Line Assembly
- Simulates Target System Memory
- Breakpoints and Single-Step Execution
- Simulates Parallel and Serial Port I/O
- Profiling of Code Execution History

C COMPILER & RUNTIME LIBRARY
- ANSI C Compliant
- Supports In-Line Assembly Code
- Incorporates Optimizing Algorithms
- Produces ROMable Code
- Floating-Point Emulation Support
- C-Callable Library with ANSI-Standard and DSP Functions
- Simplified Interrupt Handling via Library Functions

GENERAL DESCRIPTION

The ADSP-2100 Family Development Software is a complete set of software design tools that allow the programming of applications for this family of DSP microprocessors, including the ADSP-2100, ADSP-2101 (ADSP-2102), ADSP-2105 (ADSP-2106), ADSP-2111 (ADSP-2112), and ADSP-21msp50. The development system includes C and assembly language programming tools as well as processor simulators to facilitate software design and debug.

The software development system includes several programs: System Builder, Assembler, Linker, PROM Splitter and HIP Splitter, Processor Simulators and C Compiler. Release 3.0 (and higher) of the development software is available for the IBM (or IBM-compatible) PC and SUN4 workstation platforms.

Code generation is accomplished by writing C and/or assembly language source code modules. Each C and assembly module is compiled and/or assembled separately. Multiple modules are then linked together to form an executable program.

Programming in assembly language is eased by the highly readable *algebraic syntax* of the ADSP-2100 Family instruction set. A multiply-accumulate instruction, for example, is written in the same manner as the actual equation. The algebraic statement $r = r + x*y$ is coded in assembly language as $MR = MR + MX0*MY0$.

The Simulator provides reconfigurable windows that display different portions of the hardware environment. To replicate the target hardware, the simulator configures memory according to the architecture description file generated by the system builder, and simulates I/O ports according to user-entered commands. This simulation allows the user to debug the system and analyze performance before committing to a hardware prototype.

REV. A

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 617/329-4700 Fax: 617/326-8703 Twx: 710/394-6577
Telex: 924491 Cable: ANALOG NORWOODMASS

ADDS-21XX-SOFTWARE

C COMPILER

The C Compiler reads source files written in ANSI-standard C and generates ADSP-2100 Family assembly language files. The compiler allows system programmers to code applications in C or in mixed assembly and C. Applications written in C are compiled, assembled, and linked to create executable ADSP-21xx programs that can be debugged on an ADSP-21xx Simulator or Emulator.

The compiler includes features that optimize the output assembly code for length or execution speed. These features include algorithms to perform the following optimizations:

- value propagation
- dead code elimination
- base address removal
- common subexpression elimination
- loop-invariant expression removal

RUNTIME LIBRARY

The C Compiler comes with a set of C-callable library routines that ease program development. The library routines include a set of ANSI-standard functions as well as routines that perform operations commonly used in digital signal processing. The C language is especially dependent upon library functions to perform basic operations not provided by the language, for example input/output, memory allocation, and process control.

The ADSP-2100 Family Runtime Library includes the following ANSI-standard function categories:

- Standard Library
- Mathematics
- Signal Handling
- Variable Arguments
- Character Handling
- String Handling

The library also includes the following categories of signal processing functions developed by Analog Devices:

- DSP Filters
- Fast Fourier Transforms
- Matrix Operations
- Fractional Mathematics

Interrupt Handling in C

Hardware interrupts are supported in the C environment—a key feature for ease of programming. Interrupts can be processed with the signal handling functions of the C library: *signal*, *raise*, and *interrupt*. These functions are designed to process ADSP-21xx interrupts such as serial port transmit and receive interrupts, timer interrupt, and external interrupt request signals.

The *signal* and *raise* routines vector to a C interrupt service routine when a specific interrupt occurs. This allows the entire application to be written in C, without assembly language service routines. The signal handling routines save and restore registers, but this overhead is usually minimal compared to overall program execution time. If you choose to write custom interrupt service routines in assembly language, the *signal* and *raise* functions may still be used to set up the service routines in the C environment.

Signal Processing Functions

The C library includes a set of functions developed by Analog Devices that perform digital signal processing operations. Some of the key functions are:

ffir	FIR filter defined by coefficients and delay line arguments
fiir	IIR filter defined by coefficients and delay line arguments
xbiquad	cascaded biquad IIR filter
xplatt	all-pole lattice filter
xzlatt	all-zero lattice filter
xsgu	single-precision stochastic gradient update
xautocorr	autocorrelation of input signal
xcrosscorr	cross-correlation of two signals
xfft	Fast Fourier Transform
xifft	inverse Fast Fourier Transform
xmatadd	matrix addition
xmatmul	matrix multiplication
xmatsub	matrix subtraction

SYSTEM BUILDER

The system development process begins with the task of target system definition. The System Builder reads a system specification file written by the user and outputs an architecture description file that passes information about target system hardware and memory to the Linker, Simulator, and Emulator.

ASSEMBLER

The Assembler reads source files containing ADSP-2100 Family assembly language and generates a relocatable object file. The Assembler includes a preprocessor that allows the use of C preprocessor directives in assembly code, for example *#define, #include, #if, #ifdef, #else*.

Assembler directives are used to define code modules, data buffers, data variables, and memory-mapped I/O ports. Either assembler directives or C preprocessor directives can be used to define and invoke macros.

LINKER

The Linker processes separately assembled object files to create a single executable program. It assigns memory locations to code and data in accordance with the System-Builder-defined architecture file.

The Linker generates a symbol table file containing a list of all symbols encountered in the processed files; this includes variable names and program labels. The symbol table file is read by the Simulators and Emulators to allow symbolic debugging. A map listing file is also generated, which includes address maps of program memory and data memory.

PROM SPLITTER & HIP SPLITTER

The PROM Splitter translates an ADSP-21xx executable program into a file that can be used to program PROM memory devices. The PROM Splitter's output file can be generated in Motorola S Record or Intel Hex Record format. (Motorola S2 format is supported for byte stream output.)

The HIP Splitter utility is used to generate ADSP-2111 and ADSP-21msp50 programs to be downloaded from a host processor. The program is loaded via the Host Interface Port (HIP) of the ADSP-2111 or ADSP-21msp50. The HIP Splitter's output file is generated in Motorola S Record format.

SIMULATOR

A software simulator is provided for each processor of the ADSP-2100 Family. The Simulators provide an instruction-level simulation of program execution. The Simulators model system memory and I/O according to the contents of the system architecture file and provide windows to display different portions of the target system hardware. The window- and menu-based user interface allows system designers to interactively observe and alter register and memory contents, providing a powerful debug environment. Simulator commands are entered with a mouse or from the keyboard.

Features offered by the ADSP-2100 Family Simulators include:

- simulation of program and data memory
- simulation of memory-mapped I/O ports
- simulation of interrupts
- simulation of program booting from PROM or host processor
- simulation of processor serial ports
- breakpoints and break expressions
- single-step execution
- profiling of code execution patterns for program optimization

Figure 1 shows the Simulator display. (Note: The ADSP-2101 Simulator is used for simulation of ADSP-2105 systems.)

DEVELOPMENT PROCESS

Figure 2 shows the process of compiling, assembling, linking and simulating a program, indicating the input and output of each step. Hardware development tools used for software debugging, such as EZ-ICE™ Emulators, are also shown.

MINIMUM PC REQUIREMENTS

The minimum system required to use the development software consists of the following:

- 286, 386, or 486-based PC
- 640K RAM*
- EGA or VGA monitor and color video card
- high-density floppy disk drive
- DOS 3.0 or higher

*The development software supports PCs with extended memory; 2 MB RAM (total) is recommended for optimal performance.

A mouse is recommended for use with the Simulators.

EZ-ICE is a trademark of Analog Devices, Inc.

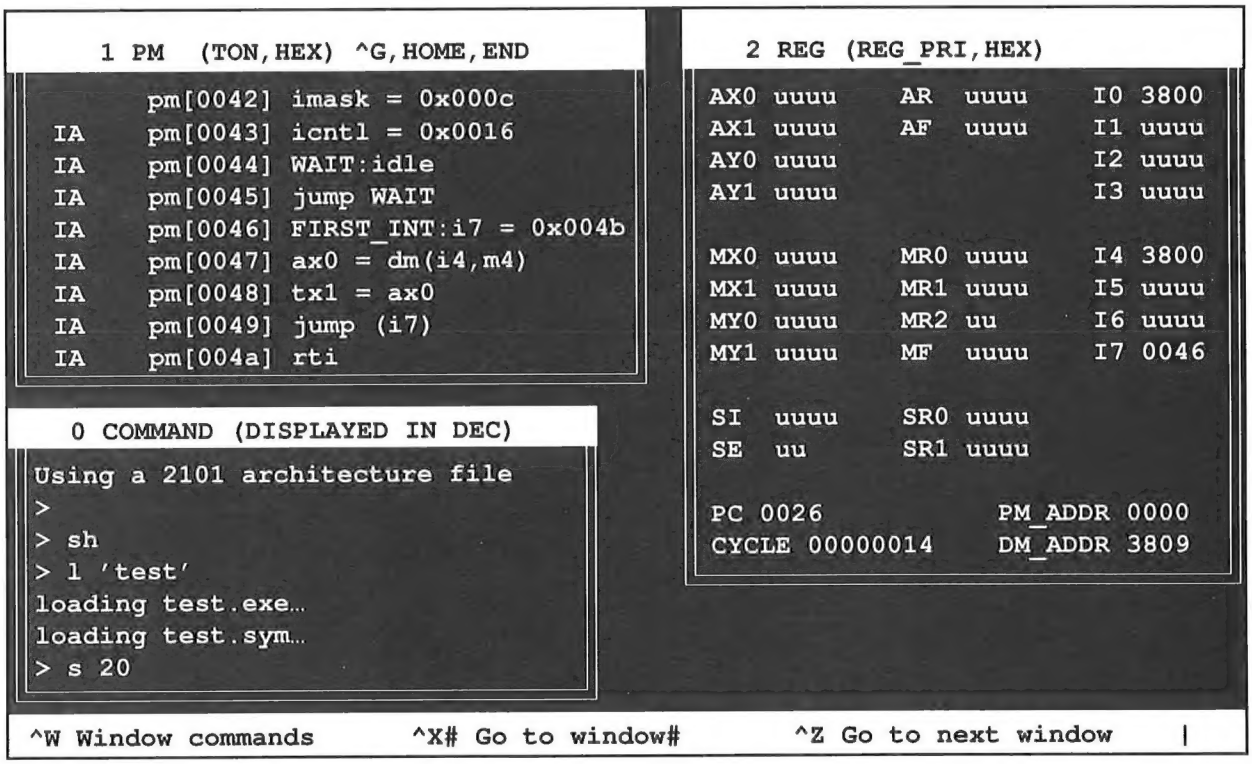

Figure 1. Simulator Display

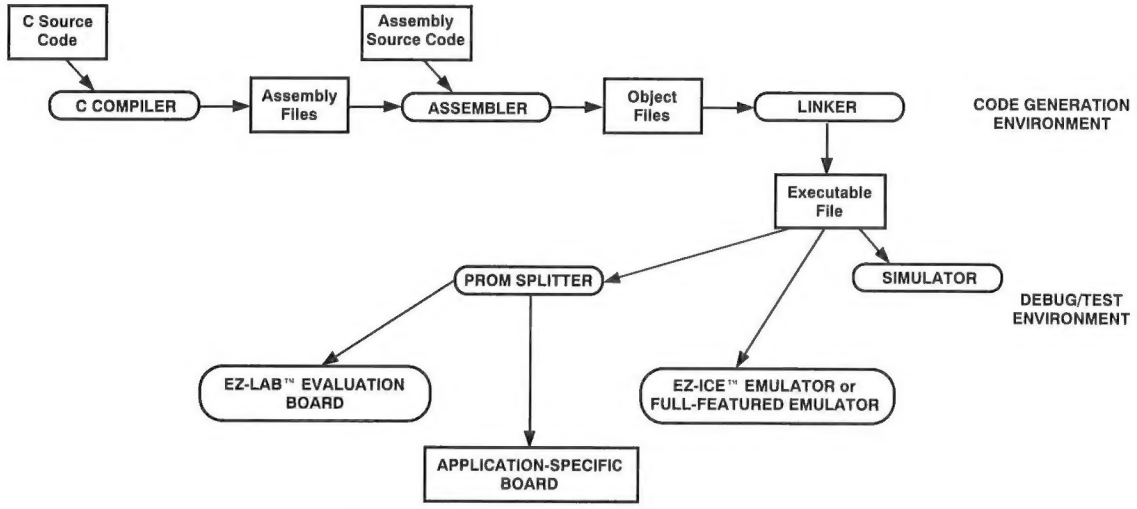

Figure 2. Software Development Process

ORDERING INFORMATION

The ADSP-2100 Family Development Software is available for IBM-compatible PC and SUN4 workstation platforms. Two sales packages are offered: the DSW package includes all ADSP-2100 Family Simulators, System Builder, Assembler, Linker, and PROM Splitter/HIP Splitter, while the BUN package includes all of these tools plus the C Compiler & Runtime Library.

ORDERING GUIDE

Part Number	Description
ADDS-21XX-DSW-PC	Simulators, System Builder, Assembler, Linker, PROM Splitter/HIP Splitter
ADDS-21XX-BUN-PC	C Compiler & Runtime Library, Simulators, System Builder, Assembler, Linker, PROM Splitter/HIP Splitter
ADDS-21XX-C-UP-PC	Upgrades owners of DSW package to BUN package (adds C Compiler & Runtime Library)
ADDS-21XX-DSW-SUN	Simulators, System Builder, Assembler, Linker, PROM Splitter/HIP Splitter
ADDS-21XX-BUN-SUN	C Compiler & Runtime Library, Simulators, System Builder, Assembler, Linker, PROM Splitter/HIP Splitter
ADDS-21XX-C-UP-SUN	Upgrades owners of DSW package to BUN package (adds C Compiler & Runtime Library)

C1495a–6–5/92

PRINTED IN U.S.A.